

Title	A Language-Based Approach to Robust Context-Aware Software( Abstract_要旨 )
Author(s)	Inoue, Hiroaki
Citation	Kyoto University (京都大学)
Issue Date	2018-03-26
URL	<a href="https://doi.org/10.14989/doctor.k21217">https://doi.org/10.14989/doctor.k21217</a>
Right	許諾条件により本文は2018-04-12に公開
Type	Thesis or Dissertation
Textversion	ETD

( 続紙 1 )

京都大学	博士（情報学）	氏名	井上 裕昭
論文 題目	A Language-Based Approach to Robust Context-Aware Software (堅牢な文脈認識ソフトウェア開発のためのプログラミング言語の研究)		
(論文内容の要旨)			
<p>本論文は、文脈認識ソフトウェアと呼ばれる、その挙動が外界の系(これを文脈と呼んでいる)に大きくかつ即応的に影響を受けるようなソフトウェアシステムを主題としている。IoTシステムや、ロボット、ファクトリーオートメーションを制御するソフトウェアは、文脈認識ソフトウェアであると考えられるが、このようなソフトウェアは、文脈の動的性や予測困難性などから、従来の計算機の中で閉じたソフトウェアに比べて、堅牢に構築することがより困難である。この問題に対し、本論文は、文脈認識ソフトウェアを記述するためのプログラミング言語を改良することにより堅牢性の向上を図った。論文は全6章から構成されている。</p> <p>第1章では、研究の背景である文脈認識ソフトウェアとその問題が議論され、その問題に対する本論文のアプローチと技術的貢献が概説されている。</p> <p>第2章では、文脈指向言語と呼ばれる、オブジェクト指向言語を文脈認識ソフトウェア向けに拡張した言語に対する型システムが提案され、言語の形式モデルを通じて、その正当性が数学的に証明されている。この正当性は、型システムによる実行前の検査を通過したプログラムは、プログラムの実行過程を表した操作的意味論においてメソッド呼び出しに失敗しないことを保証する、型健全性と呼ばれる性質である。対象となっている文脈指向言語は、クラスだけでなく層についても継承と部分型機構を持ち、また、層が第一級の値として使えるといった特徴がある。</p> <p>第3章では、文脈認識ソフトウェアが外界の異常を検知した際の例外処理を簡潔に記述するためのプログラミング言語ContextWorkflowの設計とその処理系の構築について議論が行われている。この言語におけるプログラムは、正常処理と、補償処理と呼ばれる異常時の回復処理のふたつの処理記述をペアにしたプログラム部品を組み合わせで構成され、異常検知と検知後に実行すべき補償処理の構成を自動的に行う機構が備わっている。また、プログラムの実行状態を一旦データ化、保存し、後で再開するための機能や、完了した一部の処理については補償処理を無効化する機能などによって、文脈認識ソフトウェアに要請される様々な状況に対応することが可能になっていることが例を通じて示されている。</p> <p>第4章では、文脈に依存するソフトウェアの振舞いをモジュール化するためのライブラリGearの設計について議論が行われている。このライブラリは、関数型リアクティブプログラミングと呼ばれる手法を応用し、複数のセンサーなどから得られる情報を統合して文脈情報としてシステムに自動的に提供するための機構が備わっている。さらに、文脈に依存する振舞いの記述を、関数とその関数がどういった文脈で実行されるかの条件のペアの集まりとして表現することで、細粒度な分割が可能になっている。ライブラリが提供する実行時システムは、文脈の変化を監視し、文脈の変化に応じた適切な処理を自動的に選択するための機能を提供するものとなっている。例を通じて、そのライブラリの概要とユースケースが議論されている。</p> <p>最後に、第5章で関連研究について議論がされ、第6章で本論文の結論と今後の方向性が与えられている。</p>			

注) 論文内容の要旨と論文審査の結果の要旨は1頁を38字×36行で作成し、合わせて、3,000字を標準とすること。

論文内容の要旨を英語で記入する場合は、400～1,100wordsで作成し  
審査結果の要旨は日本語500～2,000字程度で作成すること。



(続紙 2 )

(論文審査の結果の要旨)

文脈認識ソフトウェアは、文脈の動的性や予測困難性から構築することが困難であり、この構築の困難性の軽減は挑戦的な研究課題である。本研究はプログラミング言語を改良するというアプローチで、文脈認識ソフトウェア特有のエラーを実行前に検知する技術と、実行時に発生した異常に対する適切な処理を組み合わせることで、文脈認識ソフトウェアの堅牢性を向上させるという研究課題に取り組んでいる。

本論文で示された成果についての特徴は以下の通りである。

まず、層に対する継承、部分型、そして第一級の層を使うことができるようなプログラミング言語は、その振舞いが非常に複雑であり、メソッド呼び出しが必ず成功することを保証できる型システムはこれまで存在しなかった。実際、上記のような機能を持つ言語の実装は存在するが、オブジェクトに対するメソッド呼び出しが実行時に失敗してしまう例が知られている。本論文の型システムはこの問題点を克服するものであり、しかも、その正しさ、つまりメソッド呼び出しが必ず成功することを形式的なプログラミング言語のモデルを通じて数学的に証明することに成功している。先行研究では、継承や部分型がなく、層が第二級であるような言語に対する型システムの理論が研究されているが、それに比べても本論文は、型健全性の保証には部分型が二種類必要であることを示すなど、理論の大幅な洗練がみられる。この結果は理論的なものだが、既存の現実的な文脈指向言語の安全性を高めることが可能となることを示したと考えられる。

また、エラー発生時の回復処理をより簡潔に記述するための仕組み、そして、文脈依存の振舞いのモジュール化を支援するための新しい仕組みを考案している。これらは関数型リアクティブプログラミングやワークフローと呼ばれる、必ずしも文脈認識ソフトウェアとは関連の不高くない分野で提案されたプログラミングパラダイムを組み合わせることで設計されている。文脈情報管理のための関数型リアクティブプログラミングとエラー回復処理のためのワークフローというアイデアは、合理的ながらこれまでに考えられてこなかった組み合わせであり、本論文の新規性を高めている。さらに、既存のアイデアの単純な組み合わせに留まらず、文脈認識ソフトウェアに特徴的な状況に対応するための機能なども盛り込まれている点にも本論文で提案するシステムの特徴がある。

そして、これらの言語・ライブラリの提案は、単なるアイデアに留まらず、実装され、簡単な文脈認識ソフトウェアの記述例とともにオープンソースソフトウェアとして既に公開されている。

以上のように、本論文は、文脈認識ソフトウェアのためのプログラミング言語について、堅牢性の観点からの問題点を体系的に整理し、その課題に対して、エラーを実行前に検知するための型システムの理論を構築するとともに、エラーからの回復処理の簡潔な記述や文脈依存の処理のモジュール化を支援するための言語設計および実装を行った点に意義がある。さらに、文脈の概念と、既存のリアクティブプログラミングやワークフローと呼ばれるプログラミングパラダイムとを融合させることで、より堅牢な文脈認識ソフトウェアの新しい設計・実現方法につながる手法の提案になっている点にも意義がある。よって、本論文は博士（情報学）の学位論文として価値あるものと認める。平成30年2月16日に論文内容およびそれに関連した口頭試問を行った結果合格と認めた。

注) 論文審査の結果の要旨の結句には、学位論文の審査についての認定を明記すること。

更に、試問の結果の要旨（例えば「平成 年 月 日論文内容とそれに関連した口頭試問を行った結果合格と認めた。」）を付け加えること。

Webでの即日公開を希望しない場合は、以下に公開可能とする日付を記入すること。

要旨公開可能日： 年 月 日以降